Live fast and die young

Evolution and fate of massive stars

Renzo, M.

Systematic survey of the effects of wind mass loss algorithms on the evolution of single massive stars

A

A.1 Parametric Wind Algorithms

Here, we give a brief review of the physical assumptions entering in each mass loss algorithm and the corresponding prescribed rate. We summarize in Tab. A.1 the scaling of the mass loss rate with physical parameters of the stellar model for each mass loss algorithm considered in this study.

A.1.1 Vink *et al.* (V)

The wind mass loss algorithm proposed by Vink et al. (2000, 2001) is based on Monte Carlo simulations of the photon transport in the stellar atmosphere to evaluate the radiative acceleration. Their algorithm is explicitly metallicity dependent and is supposed to be applied only to single OB stars with metallicity $1/30 \leq Z/Z_{\odot} \leq 3$ and effective temperature $12\,500\ \mathrm{K} < T_{\mathrm{eff}} < 50\,000\ \mathrm{K}$, so during the hot, blue evolutionary phase. In this temperature range, a non-monotonic behavior of $\dot{M}$ as a function of T_{eff} is expected (the so-called "bi-stability jumps"): normally the lower the temperature, the lower the mass loss rate (because the radiation pressure is proportional to T^4). However, when the temperature drops below $\sim 25\,000$ K, the recombination of Fe IV $\rightarrow$ Fe III provides a new ion with an increased number of lines to drive the wind, resulting in an increased mass loss rate at lower temperatures around $T_{\mathrm{eff}} \sim 25\,000$ K. This happens with different iron ion recombinations also at $T_{\mathrm{eff}} \sim 12\,000$ K. Vink et al. (2000, 2001) provide two different formulae for above and below $T_{\mathrm{eff}} \sim 25\,000$ K,

$$\begin{aligned}\log_{10}(-\dot{M}) = &-6.697(61) + 2.194(21)\log_{10}\left(\frac{L}{10^5 L_{\odot}}\right) + \\ &-1.313(46)\log_{10}\left(\frac{M}{30M_{\odot}}\right) - 1.226(37)\log_{10}\left(\frac{v_{\infty}}{2v_{\mathrm{esc}}}\right) + \\ &+0.933(64)\log_{10}\left(\frac{T_{\mathrm{eff}}}{40\,000\ \mathrm{K}}\right) - 10.92(90)\log^2_{10}\left(\frac{T_{\mathrm{eff}}}{40\,000\ \mathrm{K}}\right) + \\ &+0.85(10)\log_{10}\left(\frac{Z}{Z_{\odot}}\right)\ ,\end{aligned} \tag{A.1}$$

Table A.1: Functional dependence of the mass loss rate $\dot{M}$ on stellar and wind parameters for each algorithm combination and mass loss phase. L is the luminosity, M is the mass, $v_{\rm esc}$ is the escape velocity, v_∞ is the final velocity of the wind, $v_{\rm th}$ is the thermal velocity, $T_{\rm eff}$ is the effective temperature, R the radius, $\Gamma_{\rm E} = L/L_{\rm Edd}$ the Eddington ratio, and X_s, Y_s, and Z_s are the surface hydrogen abundance, helium abundance, and metallicity, respectively. The scalings are obtained from the algorithms described in Appendix A.1, and errors and overall multiplying factors are omitted for the sake of brevity. See Tab. 2.2 for the naming convention for mass loss algorithm combinations.

ID	V-dJ-NL	V-NJ-NL	V-vL-NL	V-dJ-H	V-NJ-H	V-vL-H
Hot	$L^{2.210}M^{-1.339}\left(\frac{v_\infty}{2v_{\rm esc}}\right)^{-1.601}T_{\rm eff}^{1.07}Z^{0.85}$ if $T_{\rm eff} < 22.5$ kK					
	$L^{2.194}M^{-1.313}\left(\frac{v_\infty}{2v_{\rm esc}}\right)^{-1.226}Z^{0.85}T_{\rm eff}^{(0.933-10.92\log_{10}(T_{\rm eff}/40{\rm kK}))}$ if $T_{\rm eff} > 27.5$ kK					
Cool	$L^{1.769}T_{\rm eff}^{-1.676}$	$L^{1.24}M^{0.16}R^{0.81}$	$L^{1.05}T_{\rm eff}^{-6.3}$	$L^{1.769}T_{\rm eff}^{-1.676}$	$L^{1.24}M^{0.16}R^{0.81}$	$L^{1.05}T_{\rm eff}^{-6.3}$
WR	$L^{1.29}Y_s^{1.73}Z_s^{0.47}$			$L^{1.5}10^{-2.85X_s}$ if $\log_{10}(L/L_\odot) > 4.5$		
				$L^{6.8}$ if $\log_{10}(L/L_\odot) \leq 4.5$		
ID	K-dJ-NL	K-NJ-NL	K-vL-NL	K-dJ-H	K-NJ-H	K-vL-H
Hot	$L^{1.779}v_{\rm th}^{-1.169}[GM(1-\Gamma_{\rm E})]^{0.610}$					
Cool	$L^{1.769}T_{\rm eff}^{-1.676}$	$L^{1.24}M^{0.16}R^{0.81}$	$L^{1.05}T_{\rm eff}^{-6.3}$	$L^{1.769}T_{\rm eff}^{-1.676}$	$L^{1.24}M^{0.16}R^{0.81}$	$L^{1.05}T_{\rm eff}^{-6.3}$
WR	$L^{1.29}Y_s^{1.73}Z_s^{0.47}$			$L^{1.5}10^{-2.85X_s}$ if $\log_{10}(L/L_\odot) > 4.5$		
				$L^{6.8}$ if $\log_{10}(L/L_\odot) \leq 4.5$		

for 27 500 K $< T_{\rm eff} \leq$ 50 000 K, and

$$\begin{aligned}\log_{10}(-\dot{M}) = &-6.668(80) + 2.210(31)\log_{10}\left(\frac{L}{10^5 L_\odot}\right) + \\ &-1.339(68)\log_{10}\left(\frac{M}{30M_\odot}\right) - 1.601(55)\log_{10}\left(\frac{v_\infty}{2v_{\rm esc}}\right) + \\ &+1.07(10)\log_{10}\left(\frac{T_{\rm eff}}{40\,000\ {\rm K}}\right) + 0.85(10)\log_{10}\left(\frac{Z}{Z_\odot}\right)\ ,\end{aligned} \tag{A.2}$$

for 12500 K $< T_{\rm eff} \leq$ 22500 K. In between these two temperature ranges, it is common practice to simply interpolate between the two formulae (cf. Appendix A.2). The numbers in parenthesis are the estimates of the error on the last digit reported, according to Vink et al. (2001). These error estimates are usually neglected in stellar evolutionary calculations.

A.1.2 Kudritzki *et al.* (K)

The mass loss algorithm proposed by Kudritzki et al. (1989) is based on an analytic solution of the equations for a stationary, isothermal, spherically-symmetric, ideal (neither viscosity nor heat conduction) gas flow with no magnetic fields and no rotation. They include the radiative acceleration $g_{\rm ph}$ in the momentum equation using the standard parametrization of Castor et al. (1975) (see also Eq. 16 in Pauldrach et al. 1986),

$$g_{\rm ph} = g_{\rm rad}^{\rm Th}\left(1 + k\left(\frac{\sigma_{\rm Th} c_s}{dv/dr}\right)^{-\alpha}\left(\frac{2n_e}{\sqrt{1-(1-(R/r)^2}}\right)^{\delta} {\rm CF}\left(r, v, \frac{dv}{dr}, \alpha\right)\right) \ , \tag{A.3}$$

where $g_{\rm rad}^{\rm Th}$ is the radiative acceleration due to Thomson scattering, and the second term in the brackets is the so-called "force multiplier", that is the line acceleration in units of $g_{\rm rad}^{\rm Th}$. It depends on the Thomson cross section $\sigma_{\rm Th}$, the speed of sound c_s, the electron number density n_e, and three free parameters k, α, and δ. k can be interpreted roughly as the number of lines strong enough to have an effect, and α as the slope of the distribution of the number of lines as a function of their strength. The parameter δ and the correction factor CF, which is the ratio between the opacity as a function of the incoming angle and the opacity in the radial direction, are used to include the "finite cone-angle effect" to account for photons travelling in non-radial directions (e.g., Castor et al. 1975). If δ and CF were both equal to one, the parametrization would be valid only in the "radial streaming" limit (Abbott 1982; Pauldrach et al. 1986), that is considering only incoming photons from the radial direction. While this is a good approximation in the outer portion of the wind, where $r \gg R$ and R is the stellar radius, it is quite poor in the inner portion, where the mass loss rate is determined and photons traveling in non-radial directions can have a relevant effect. For completeness, we report the expression for the correction factor CF as a function of the dimensionless radial coordinate $x \stackrel{\rm def}{=} r/R$ and $h \stackrel{\rm def}{=} d\log_{10}(x)/d\log_{10}(v)$, cf. Eq. (4) in Kudritzki et al. (1989),

$${\rm CF}\left(r, v, \frac{dv}{dr}, \alpha\right) = \frac{1}{\alpha+1}\frac{x^2}{1-h}\left(1 - \left(1 - \frac{1}{x^2} + \frac{h}{x^2}\right)^{\alpha+1}\right) \ . \tag{A.4}$$

We refer the reader to Castor et al. (1975); Pauldrach et al. (1986); Vink (2015) and references therein for more details on this parametrization.

To find an analytic solution to their model, further assumptions are needed. Kudritzki et al. (1989) impose a "β-law" velocity field (a common assumption found to be close to numerical solutions, see Lamers 2013),

$$v(r) = v_\infty\left(1 - \frac{R}{r}\right)^{\beta} \ , \tag{A.5}$$

where β is a free parameter assumed to be $\beta = 1$, v_∞ is the asymptotic velocity of the wind and R is the stellar radius. This assumption means Kudritzki et al. (1989) do not solve for the dynamics of the system, but rather assume the velocity structure and solve self-consistently for the density and the acceleration. In the limit where $v \gg c_s$ (reasonable in the outer portion of the wind), Kudritzki et al. (1989) find a solution of the form

$$\begin{aligned} \dot{M} \equiv \dot{M}(\alpha, \delta, k, M, L, v_{th}) = \tilde{D}(\alpha, \delta, v) \left(\frac{\sigma_{\rm Th} k L}{4\pi c} \right)^{1/(\alpha-\delta)} \times \\ \times \left(\frac{4\pi\alpha}{\sigma_{\rm Th} v_{\rm th}} \right)^{\alpha/(\alpha-\delta)} \left(\frac{1-\alpha}{GM(1-\Gamma)} \right)^{(1-\alpha)/(\alpha-\delta)} , \end{aligned} \tag{A.6}$$

where $v_{\rm th}$ is the thermal velocity of protons, k, α, and δ are the free parameters, L is the luminosity, $v_{\rm th}$ is the thermal velocity, $\Gamma = L/L_{\rm Edd}$ is the Eddington ratio, and $\tilde{D}$ is a function of the free parameters and the velocity (cf. Eqs. 47, 62, 65 in Kudritzki et al. 1989).

The main limitation of Eq. A.6 is that the parameters k, α and δ are not constants, but rather depend on the optical depth. The numerical values commonly adopted are $\alpha = 0.657$, $\beta = 1$, $\delta = 0.095$, and $k = 0.085$ and they are calibrated on ζ Puppis by Pauldrach et al. (1994). These should be interpreted as values averaged over the optical depth.

A.1.3 de Jager *et al.* (dJ)

The wind algorithm proposed by de Jager et al. (1988) is an empirical relationship of the form $\dot{M} \equiv \dot{M}(T_{\rm eff}, L)$. This choice of variables uses only observable quantities, making it easy to track the mass loss rate while the star moves on the HR diagram. This allows a better understanding of how mass loss changes during stellar evolution. The drawback is that no information about the physical mechanism driving the wind is considered.

To formulate a reliable mass loss algorithm, de Jager et al. (1988) collect from the literature mass loss rates observed with different techniques for a sample of galactic stars with spectral types from O to M. They determine the "average" measured mass loss rate for stars observed with multiple techniques and the deviation from this average for each available measurement and for each star. Then, they define the "average intrinsic error per determination" as the one-sigma value of the distribution of these deviations. They fit their entire data sample with a sum of Chebychev polynomials of the first kind $T_n(x) = \cos(n \arccos(x))$:

$$\log_{10}(-\dot{M}) = \sum_{n=0}^{N} \sum_{\substack{i=0 \\ j=n-i}}^{i=n} a_{ij} T_i(\log_{10}(T_{\rm eff})) \cdot T_j(\log_{10}(L)) \ . \tag{A.7}$$

What is commonly used in stellar evolution codes is the first-order approximation to Eq. A.7,

$$\log_{10}(-\dot{M}) = 1.769 \log_{10}(L/L_\odot) - 1.676 \log_{10}(T_{\rm eff}/[\rm K]) - 8.158 \ . \tag{A.8}$$

To asses the quality of this fit, they derive the distribution of the differences between the observed values of $\log_{10}(-\dot{M})$ with the result of Eq. A.8. The standard deviation of this distribution is ~ 0.45, slightly larger than the "averaged intrinsic error per determination" previously determined. This indicates that quantities other than $T_{\rm eff}$ and L must physically enter in the determination of $\dot{M}$, and the parametrization in Eq. A.8 is incomplete. The main limitation of the dJ algorithm is that it is representative of the "averaged statistical behavior" of stellar winds in the entire HR diagram, which might average over different physical regimes. It is important to note that WR and Be stars are intentionally excluded from the data sample used to derive Eq. A.8.

A.1.4 Nieuwenhuijzen & de Jager (NJ)

The algorithm of Nieuwenhuijzen & de Jager (1990) is intended as an improvement over the de Jager et al. (1988) algorithm of Eq. A.8, since it is derived from the same data sample with a similar method but includes the dependence of mass loss on the total stellar mass. The goal of this is to capture one of the missing stellar parameters entering in the mass loss determination indicated by the large standard deviation of Eq. A.8. The Nieuwenhuijzen & de Jager (1990) algorithm also translates the temperature dependence into a radius dependence.

Since the total mass is not a directly observable quantity for single stars, the authors' mass determination is based on stellar model calculations. The theoretical models used are from Maeder & Meynet (1988, 1989). However, different stellar evolution codes consider a large variety of physical processes (e.g., for mixing, mass loss, etc.), or just use different implementations of them. Hence, there is spread in the stellar masses found at the same point of an evolutionary track. This implies that the Nieuwenhuijzen & de Jager (1990) mass loss algorithm depends on the set of stellar models used to derive it. This drawback applies to all mass loss algorithms involving a functional dependence of the form $\dot{M} \equiv \dot{M}(M)$ derived from stellar evolution calculations. However, Mauron & Josselin (2011) suggests that the dependence on the total mass of the Nieuwenhuijzen & de Jager (1990) mass loss rate is so weak (see Eq. A.11) that it can be averaged (by substituting M, which changes during the evolution, with a constant value) without dramatic consequences on the evolved stellar model.

Stars with different masses pass through the same point on the HR diagram at different stages of their evolution. Therefore, in order to include the total mass as a variable for the mass loss rate, Nieuwenhuijzen & de Jager (1990) determine an "average expected mass" $\bar{M}$ of a star at a given $(T_{\rm eff}, L)$ point. This value is derived as follows. The authors define a "dwell time" representing the time for a star to travel over a unit length track on the HR diagram,

$$t^{(d)} \stackrel{\rm def}{=} \frac{\delta t}{\sqrt{[\delta \log_{10}(T_{\rm eff}/[\mathrm{K}])]^2 + [\delta \log_{10}(L/L_\odot)]^2}} \ , \tag{A.9}$$

where δt is the time spent to travel over the $(\delta \log_{10}(T_{\rm eff}/[\mathrm{K}]), \delta \log_{10}(L/L_\odot))$ distance.

For every point on the HR diagram, there are N left-ward or right-ward subtracks of the

stellar evolutionary tracks crossing it. Let $t_n^{(d)}$ be the dwell time for the n-th subtrack. The average expected mass $\bar{M}$ is obtained as:

$$\bar{M} = \frac{\sum_{n=1}^{N} \Psi(M_n) \frac{dM_n}{d\log_{10}(L/L_\odot)} t_n^{(d)} M_n}{\sum_{n=1}^{N} \Psi(M_n) \frac{dM_n}{d\log_{10}(L/L_\odot)} t_n^{(d)}} \quad , \tag{A.10}$$

where Ψ is the initial mass function for stars on the subtrack considered, and $dM_n/d\log_{10}(L/L_\odot)$ is the density of tracks over a unit $\log_{10}(L/L_\odot)$ interval.

The authors perform a fit of the data set[1] used in de Jager et al. (1988), adding the value of $\bar{M}$ from Eq. A.10 to the set, and they find the interpolation formula[2] (where $\bar{M}$ is substitued by the total mass M to use the formula in a stellar evolution simulation)

$$\begin{aligned} \log_{10}(-\dot{M}) = &-14.02 + 1.24\log_{10}(L/L_\odot) + \\ &+0.16\log_{10}(M/M_\odot) + 0.81\log_{10}(R/R_\odot) \quad . \end{aligned} \tag{A.11}$$

With the inclusion of M among the parameters determining $\dot{M}$, Nieuwenhuijzen & de Jager (1990) obtain a standard deviation of ~ 0.37 for the distribution of the differences between the prediction of Eq. A.11 and the observed values, comparable to the standard deviation of the distribution of the differences between individual mass loss determination used as input data. The main limitations of this algorithm are its dependence on the input stellar models, and its statistical-average nature (in the same sense as de Jager et al. 1988).

A.1.5 Van Loon *et al.* (vL)

The mass loss rate of van Loon et al. (2005) is empirically determined on the basis of observations of a sample of oxygen-rich AGB and RSG stars in the Large Magellanic Cloud (LMC). The van Loon et al. (2005) analysis is based on a dust-driven wind model: AGB and RSG stars have very extended and cool envelopes where dust grains might form through sublimation. The photons from the radiation field transfer momentum to these grains, pushing them away. The dust grains drag the gas with them through collisional coupling. To obtain their mass loss algorithm, van Loon et al. (2005) fit the observed IR spectra to synthetic spectra obtained with a simple model of the gas/dust mixture (identical grains and dust-to-gas ratio set to the value observed at $Z_\odot$ rescaled to $Z_{\rm LMC}$), using $T_{\rm eff}$ and L as variables. They obtain the relationship

$$\begin{aligned} \log_{10}(-\dot{M}) = &-5.65(15) + 1.05(14)\log_{10}(L/10^4 L_\odot) + \\ &-6.3(1.2)\log_{10}(T_{\rm eff}/3\,500\,{\rm K}) \quad , \end{aligned} \tag{A.12}$$

where the numbers in parentheses indicate the estimated errors on the last digits. These errors are typically neglected in stellar evolution codes. The main limitation of this algorithm is the

[1] Which excludes WR and Be stars.

[2] The formula in the abstract of Nieuwenhuijzen & de Jager (1990) has a typo, see their Eq. 2.

high uncertainty of the dust grain properties (mass fraction, opacity, when they form, etc.). Since it gives such a high mass loss rate, the van Loon et al. (2005) algorithm is sometimes combined with a high efficiency factor to *ad hoc* mimic non-wind mass loss phenomena (e.g., Meynet et al. 2015).

A.1.6 Nugis & Lamers (NL)

The mass loss algorithm derived by Nugis & Lamers (2000) applies only to WR stars. The wind mass loss rate of these stars depends strongly on their chemical composition: not only the metallicity has an important role, but also the helium mass fraction Y. This is because the amount of helium in the stellar atmosphere influences its temperature and therefore the ionization fraction and the level populations of all other atoms and ions. Nugis & Lamers (2000) derive a mass loss rate algorithm as a function of the luminosity and the chemical composition starting from a relevant sample of observed galactic WR stars. Their sample is made of two subsets of stars: one in which both mass and distance (i.e., luminosity) are known, thanks to binarity and membership association in open clusters; and another subset for which the intrinsic luminosity is not known. They use stars from the first subset to derive an empirical bolometric correction[3]. They then use a theoretical mass-luminosity relation to infer the luminosity of stars in the second subset, and correct it with the previously derived bolometric correction. We note that the mass-luminosity relation is determined using as input the age of the star and its spectral type, not its luminosity, therefore this relation can be consistently used to estimate the luminosity of stars in the second subset (Nugis & Lamers 2000).

The mass loss rate observed for the stars in the sample are then fitted as follows. The authors make two independent fits for stars of different composition and then merge them together in a single formula, valid for all WR stars:

$$\begin{aligned}\log_{10}(-\dot{M}) = -11.0 + 1.29(14)\log_{10}(L/L_{\odot}) + \\ 1.73(42)\log_{10}(Y) + 0.47(09)\log_{10}(Z)\ ,\end{aligned} \tag{A.13}$$

where the numbers in parentheses are the estimates of the error on the last digits reported. According to Nugis & Lamers (2000), the mass loss algorithm for WR stars cannot be expressed as a function of $T_{\rm eff}$ and/or the radius of the star R. This is because WR winds are so strong (i.e., dense) that they are optically thick, and thus the observed radius is a function of the wavelength: the black body relation $L = 4\pi R^2 \sigma T_{\rm eff}^4$, where σ is the Stefan-Boltzmann constant, loses its meaning.

[3]The bolometric correction is the difference between the bolometric magnitude and the observed (visual) magnitude, influenced by the instrumental intrinsic band pass, $\mathrm{BC} \stackrel{\mathrm{def}}{=} M - M_{\rm obs}$

A.1.7 Hamann *et al.* (H)

This wind scheme, which applies only to WR stars, is a combination of the algorithms from Hamann et al. (1982), Hamann et al. (1995), and Hamann & Koesterke (1998). It is derived from a spherically-symmetric, homogeneous, and stationary (but not static, i.e., $\partial_t = 0$ but $v \neq 0$) expanding WR atmosphere model. The authors assume an *ad-hoc* velocity structure $v \equiv v(r)$ as follows. For the supersonic part of the wind, they assume a β-law of the form of Eq. A.5 with $\beta = 1$, while for the subsonic part, $v(r)$ is chosen in such way that the density approaches the hydrostatic limit. These assumptions are in reasonable agreement with the numerical solutions. Since the velocity field is imposed, the acceleration is not computed. This allows the authors to adopt a very simple chemical composition, since they do not need to evaluate the line-driven acceleration and do not need to keep track of all atomic/ionic species and their level populations. The authors include only ions of H and He, and the radiation field is considered only to determine non-LTE populations of these species. The temperature stratification is derived with the assumption of a gray LTE model, assuming a value of $T_{\rm eff}$ at the base of the atmosphere determined by the stellar luminosity L and the radius R via a black body relation $L = 4\pi R^2 \sigma T_{\rm eff}^4$, in contrast to the suggestion of Nugis & Lamers (2000). A synthetic spectrum is derived from the simulations and a best fit to the observed line profiles for the ions of H and He is obtained via variation of the stellar parameters (i.e., the radius of the inner boundary of the atmosphere R and the luminosity L, the surface hydrogen mass fraction X_s and $\dot{M}$, Hamann et al. 1995). Once the stellar parameters are known from this fit, a mass loss formula is derived for high luminosity WR stars, i.e., $\log_{10}(L/L_\odot) > 4.5$. The algorithm for the low luminosity WR stars, i.e., $\log_{10}(L/L_\odot) < 4.5$, is derived with a similar technique in Hamann et al. (1982), but the spectra fitted are from a small sample of Helium stars (i.e., stars undergoing He shell burning with most of the mass in a CO core, without H lines in their spectra). The resulting formula is

$$\log_{10}(-\dot{M}) = \begin{cases} -12.25 + 1.5\log_{10}\left(\frac{L}{L_\odot}\right) - 2.85X_s & \text{if } L > 4.5L_\odot, \\ -35.8 + 6.8\log_{10}\left(\frac{L}{L_\odot}\right) & \text{if } L \leq 4.5L_\odot. \end{cases} \tag{A.14}$$

Improvements that take into account inhomogeneities in the wind are suggested in Hamann & Koesterke (1998). Specifically, the authors suggest to reduce the wind efficiency by a factor between 2 and 3 to account for the wind clumpiness, which strongly affects the fitted spectral lines.

A.2 Input physics, customization, and resolution study

Here, we provide and discuss the `MESA` parameters not directly related to the physics of mass loss that we omit in Sec. 2.2 for brevity.

We employ the Ledoux criterion (Ledoux 1947) to determine convective stability. This means that we consider the effects of the temperature and chemical composition gradients on the stability of a region, and allow for semiconvective mixing driven by compositional

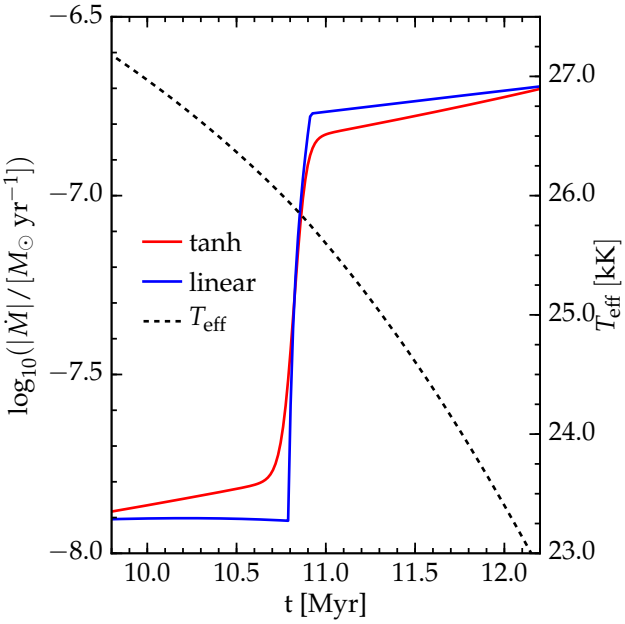

Fig. A.1: Time evolution of the mass loss rate around the bistability jump of the Vink et al. (2000, 2001) (V) mass loss rate (Sec. A.1.1) in a $M = 15M_\odot$, $\eta = 1.0$ model at $Z = Z_\odot$. The black dashed line is the effective temperature $T_{\rm eff}$, reached near the terminal age main sequence (TAMS, $X_c < 0.01$) in these simulations. The blue curve corresponds to the mass loss rate from the default `MESA` routine, which uses a linear interpolation between Eq. A.1 and Eq. A.2. The red curve is the mass loss rate from our customized routine, where we interpolate using a hyperbolic tangent.

gradients. We follow the suggestions of Sukhbold & Woosley (2014) for the treatment of convective mixing (relying on mixing length theory; Böhm-Vitense 1958) and overshooting, but we also include thermohaline mixing using the default `MESA` treatment of this process (Kippenhahn et al. 1980). Our specific `MESA` parameter choices are as follows: mixing length parameter $\alpha_{\rm mlt} = 2$, exponential overshooting/undershooting parameters $f_{\rm ov} = 0.025$ and $f_0 = 0.05$ for all regions (see Moravveji et al. 2016; Paxton et al. 2011), semiconvection efficiency of 0.2, and thermohaline mixing with coefficient 2 according to Kippenhahn et al. (1980).

We employ the 45-isotope nuclear reaction network `mesa_45.net` (in `$MESA_DIR/data/net_data/nets/`) for all models until oxygen depletion. We switch to a customized 203-isotope network for models evolved further, and initialize to zero the abundances of all the new isotopes introduced (i.e., `adjust_ abundances_for_new_isos = .false.`). We obtain the 203-isotope network by taking the union of the isotope sets of `mesa_45.net` and `mesa_201.net`.

The list of isotopes included in the network is available at https://stellarcollapse.org/renzo2017.

Stars more massive than $\sim 20\,M_\odot$ can develop a radiation-dominated, near super-Eddington

convective envelope (e.g., because of the iron opacity bump) in which the convective flux is not sufficient to transport energy outward. This can lead to numerical (and possibly physical, e.g., Langer 1997) density and pressure inversions and consequently dynamical instabilities (see, e.g., Joss et al. 1973; Paxton et al. 2013, and references therein). To handle these envelope issues, `MESA` introduces the so-called MLT++ scheme (see Paxton et al. 2013). MLT++ arbitrarily decreases the superadiabaticity in near-to-Eddington ($L \geq 0.5 L_{\rm Edd}$) radiation dominated ($P_{\rm gas}/P_{\rm tot} \leq 0.35$) convective envelopes in order to prevent numerical issues. We note that MLT++ can modify the radius and luminosity of the star by artificially enhancing the energy flux carried by convection, and thus indirectly, it can also change the mass loss rate. Furthermore, we find that MLT++ combined with the `simple_atmosphere` option for the outer boundary condition causes an unphysical oscillation of the mass loss rate and surface characteristics (e.g., luminosity L, effective temperature $T_{\rm eff}$, and radius R; Renzo 2015). To avoid this issue, we employ the more realistic atmosphere boundary condition provided by the `Eddington_grey` option in all our models. This uses the Eddington gray $T(\tau)$ relation,

$$T^4 = \frac{3}{4} T_{\rm eff}^4 \left(\tau + \frac{2}{3} \right) \ , \tag{A.15}$$

to find the boundary pressure on the photosphere, instead of guessing the approximate photosphere location.

At the beginning of each evolutionary step, `MESA` evaluates the mass loss rate $\dot{M}$ according to the algorithm specified by the user, and then removes from the surface the amount of mass $\dot{M} \times \Delta t$, where Δt is the timestep. The mass loss algorithm to evaluate $\dot{M}$ can be chosen from the many built-in algorithms, or it can be implemented by the user using the `run_star_extras.f` hooks to override the default MESA routines[4]. We use the latter option to combine three different algorithms (one for the hot phase of the evolution, one for the cool phase, and one for the WR phase, see Sec. 2.2.2). When possible, we call the built-in MESA mass loss routines from our `run_star_extras.f`, except for the V algorithm. The default implementation of the V mass loss algorithm in `MESA` uses a linear interpolation between Eq. A.1 and Eq. A.2. This results in a jump of $\dot{M}$ (as a function of time t) with discontinuous derivative (i.e., $M(t)$ is not $\mathcal{C}^2$) as $T_{\rm eff}$ drops below 25 000 K during the evolution. For a physically more realistic, smoother time dependence of $\dot{M}$, we implement a hyperbolic tangent interpolation between the two formulae. This is depicted in Fig. A.1.

To save `MESA` "`photos`"[5] and to call the built-in implementation of the K algorithm from `run_star_extras.f`, we copy the Fortran modules `write_model.mod` and `kuma.mod` from `$MESA_DIR/ star/make` into `$MESA_DIR/include` in our standard `MESA` installation. This is necessary since these two modules do not have a "public" interface (Paxton et al. 2011) in MESA release version 7624.

We describe the settings for spatial and temporal resultion in the next Sec. A.2.1. We

[4] See also http://mesa.sourceforge.net/run_star_extras.html

[5] These are binary files that allow one to restart a run and obtain bit-to-bit identical results (provided that the parameter set does not change). See also http://mesa.sourceforge.net/.

use the default MESA settings for massive stars for anything else not explicitly mentioned (see $MESA_DIR/star/ inlist_massive_defaults).

A.2.1 Resolution Dependence

Any computational study in astrophysics must carefully assess the effects of numerical resolution (in both space and time) on its results. To study the sensitivity of our results to variations in temporal and spatial resolution, we carry out a resolution study using a $30M_\odot$ star since stars of this $M_{\rm ZAMS}$ appeared to be the most sensitive to the resolution in our preliminary calculations using the default MESA parameters. The post core carbon depletion evolutionary tracks produced by the default MESA parameters show large amplitude oscillations (e.g., in the central temperature–central density plane) when varying the spatial discretization. It is unlikely that nature would do this. Thus, such oscillations are most likely artificial, and they are generally worse at higher initial mass. The aim of this section is to find a set of parameters that reduces and possibly eliminates these oscillations.

For each set of resolution parameters, we run our test model to oxygen depletion with two different mesh refinement parameters (mesh_delta_coeff=1.0 and mesh_delta_coeff = 0.5), and two different wind mass loss algorithm combinations (V-dJ and K-NJ; both with $\eta = 1.0$). The use of two different mass loss algorithm combinations allows us to check that our settings are not cherry-picked for a particular model in our grid. However, the results of our study indicate that the resolution dependence is insensitive to the mass loss algorithm combination.

To obtain numerically converged results, we use both the available MESA controls (specified in the inlists) and customized routines in run_star_extras.f.
Both are available at https://stellarcollapse.org/renzo2017.

Timestep selection throughout the evolution

To avoid overstepping relevant physical processes[6], we enforce a customized timestep control for the evolution of our models (see the routine extras_finish_step in our run_star_extras.f). In particular, we tighten the default MESA (release version 7624) timestep controls by manually enforcing

$$\Delta t_{n+1} \leq \min\{t_{\rm KH}, t_{\dot{M}}\} \ , \tag{A.16}$$

where Δt_{n+1} is the timestep proposed at the end of the $n-$ th step for the next $((n+1)-$ th) step, and all the timescales on the right hand side refer to the $n-$ th step. On the right hand side of condition A.16,

$$t_{\rm KH} \stackrel{\rm def}{=} \frac{3}{4}\frac{GM^2}{RL} \quad , \quad t_{\dot{M}} \stackrel{\rm def}{=} \frac{M}{|\dot{M}|} \ , \tag{A.17}$$

[6]Preliminary calculations with low resolution showed that in some rare cases, e.g., during the Herzsprung gap, MESA would try to overstep the Kelvin-Helmholtz timescale.

are the Kelvin-Helmholtz timescale and the mass change timescale, respectively.

The right-hand side of condition A.16 is evaluated with the quantities of the $n-$ th step, but it limits the $(n+1)-$ th step. Moreover, until oxygen depletion, we limit the timestep by setting `varcontrol_target = 1.0d-4` as the maximum relative variation for quantities in each computational cell. We also limit the amount of matter nuclearly processed in one single timestep by setting the maximum variation of the mass fraction of each element due to nuclear buring in each time step to `dX_nuc_drop_limit = 1.0d-4`.

Spatial meshing until oxygen depletion

Until oxygen depletion, we impose a maximum value for the fraction of the total mass in each computational cell by setting `max_dq = 0.5d-4` in the `inlist`. This means that each of our models has at least $1/\texttt{max\_dq} = 2 \times 10^4$ computational cells. Moreover, we refine the mesh around some specific regions of the star to focus resolution there:

- **Regions with steep temperature gradients:** to better resolve the deep interior of the star, we limit the maximum variation of $d\log_{10}(T)/dm$ across adjacent cells, where T is the temperature and m is the mass coordinate. Specifically, we impose a maximum variation of `mesh_delta_coeff`$/10$ for $\log_{10}(d\log_{10}(T)/dm + 1)$. This is is achieved using the `other_mesh_fcns_data` routine in `run_star_extras.f`;

- **Stellar surface**: to properly resolve the amount of mass lost at each timestep, we impose that the outermost $0.5\,M_\odot$ of the star is sampled by at least $500 \times$ (`mesh_delta_coeff`$)^{-1}$ cells (see `other_mesh_fcns_data` routine in `run_star_extras.f` for the implementation). We note that for each timestep Δt, $\dot{M} \times \Delta t \ll 0.5 M_\odot$;

- **Edges of burning regions**: to resolve the edges of burning regions, we constrain the spatial variation of $d\log_{10}(\varepsilon_{\rm nuc})/d\log_{10}(P)$ by multiplying its maximum allowed variation (regulated by `mesh_delta_ coeff`) by 0.015 (as in Dessart et al. 2013, private communication). This is done separately for each nuclear burning process, see `mesh_dlog_*_dlogP_extra` in the `inlists`;

- **Edges of the cores of different composition**: to resolve sharp variations in the chemical composition, we impose `mesh_delta_coeff`$/20$ as the maximum spatial variation allowed for the mass fractions of several isotopes (^{1}H, ^{4}He, ^{12}C, ^{16}O, ^{20}Ne, ^{28}Si,^{24}Mg, ^{32}S, ^{54}Fe, ^{56}Fe). These are specified via `xa_function_*` in the `inlists`.

Our models with these settings and `mesh_delta_coeff` $= 1.0$ have typically between $\sim 50\,000$ and $\sim 100\,000$ mesh points. Figure A.2 illustrates for the model using the V-dJ-NL mass loss algorithm combination and $\eta = 1.0$ that our setup does not produce any appreciable variations when changing `mesh_delta_ coeff` by a factor of 2 (corresponding to an increase of the number of spatial mesh points from 54 814 to 77 400 at oxygen depletion). We note the linear scale for the central temperature – central density evolutionary tracks.

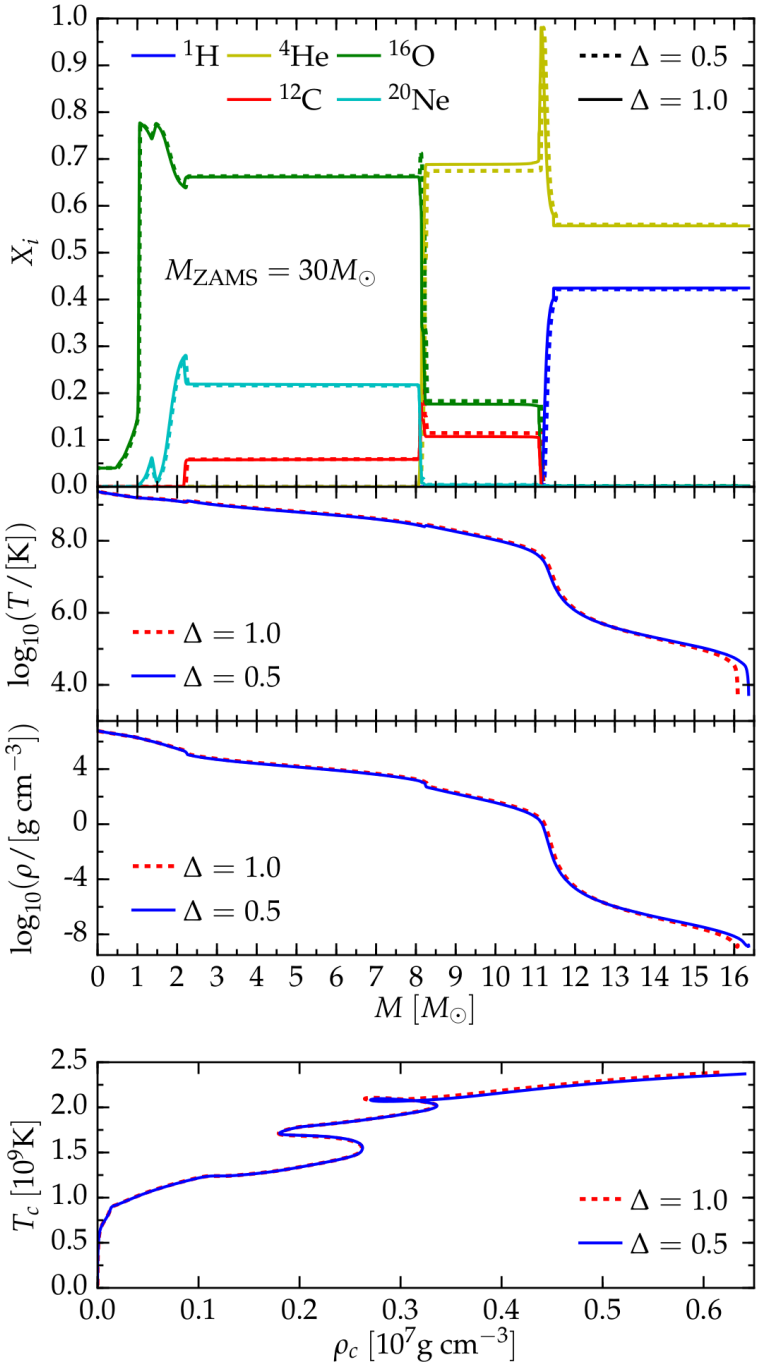

Fig. A.2: Mass fractions, temperature, and density profiles (top three panels) at oxygen depletion and central temperature – central density evolutionary track (bottom panel) from ZAMS to core oxygen depletion for the $M_{\rm ZAMS} = 30 M_\odot$ test case with the V-dJ algorithm and $\eta = 1.0$. We run the test case at the standard spatial resolution ($\Delta \equiv$ `mesh_delta_coeff`=1.0, dashed lines) and twice that resolution ($\Delta \equiv$ `mesh_delta_coeff`=0.5, solid lines).

Re-meshing after Oxygen Depletion

After oxygen depletion, we switch from a 45-isotope to a 203-isotope nuclear reaction network (cf. Sec. A.2). This switch forces us to decrease the number of spatial mesh points for the following reason: `MESA` solves the fully coupled set of equations for stellar structure and evolution Paxton et al. (2011). This results in the use of a work array of length $\mathcal{L}$ which scales as

$$\mathcal{L} \sim ((N_{\rm iso} + 5) \cdot N_{\rm z}) \cdot (3N_{\rm iso} + 9) \ , \tag{A.18}$$

where $N_{\rm iso}$ is number of isotopes in the nuclear reaction network and $N_{\rm z}$ is the number of mesh points (see the routine `get_newton_work_sizes` in `$MESA_DIR/star/private/star_newton.f90`). $\mathcal{L}$ is stored as a 4-byte integer in MESA, which sets the maximum adressable memory to ~ 17 GB (R. Farmer, private communication). Therefore, for a nuclear reaction network including ~ 200 isotopes, the maximum number of mesh points cannot exceed $\sim 17\,000$. Changing the relevant variables to an 8-byte integer data type would require substantial changes throughout MESA, which we have opted to defer to future work.

Another significant limitation on the resolution that can be achieved in the very late evolutionary phases is the stability of stellar evolution codes: many highly uncertain and/or poorly understood physical phenomena take place in the cores of evolved massive stars, and the evolutionary timescale gets progressively closer to the (neutrino-)thermal timescale and, finally, the dynamical timescale.

Experiments with the highest achievable resolution resulted in frequent failures of the code to find solutions to the stellar structure equations. However, note that by the time core oxygen burning is complete, the variations in core structure caused by different mass loss algorithm combinations are already largely developed and will be amplified by the subsequent evolution.

We reduce the number of mesh points in our models down to a few thousand (the precise value varies from model to model). We do this by restarting from oxygen depletion and running MESA's re-meshing algorithm for 100 timesteps of $\Delta t < 10^{-9}$ s with nuclear burning and neutrino cooling turned off (see `inlist_remesh`). At the same time, we shut down thermohaline mixing, which is a secular process that does not have time to produce any physically relevant change in the star after core oxygen burning. We also drop the spatial mesh refinement criteria described above for the remaining evolution. After the 100 re-meshing timesteps during which the star is *de facto* frozen in its state, we resume the evolution by again turning on nuclear burning and neutrino cooling and not imposing a maximum timestep, but we do not re-enable thermohaline mixing. We compare pre-re-meshing and post-re-meshing core structure, thermodynamics, and compactness parameter and ensure that the post-re-meshing core is still very well resolved with the reduced resolution as we proceed in the evolution toward core collapse.

As the temperature in the core increases, the nuclear burning rate accelerates. When the central temperature rises above 3×10^9 K, we progressively relax the constraints on the timestep from nuclear burning. We do this by modifying, at the beginning of each

timestep, the parameters loaded in our `inlist` using the routine `extras_startup` of our `run_star_extras.f`. Specifically, for $3.0 \leq T_c/[10^9\mathrm{K}] \leq 3.5$ we impose `dX_nuc_drop = 5d-3`, and for even higher T_c we only require `dX_nuc_drop = 5d-2` (cf. Sec. A.2.1). After silicon core depletion ($X_c(^{28}\mathrm{Si}) \leq 0.001$), we also decrease the spatial resolution of the innermost infalling core by increasing `mesh_delta_coeff_for_highT=3.0` (used where $\log_{10}(T/[\mathrm{K}]) \geq 9.3$, the value used during the previous evolution is 1.0).

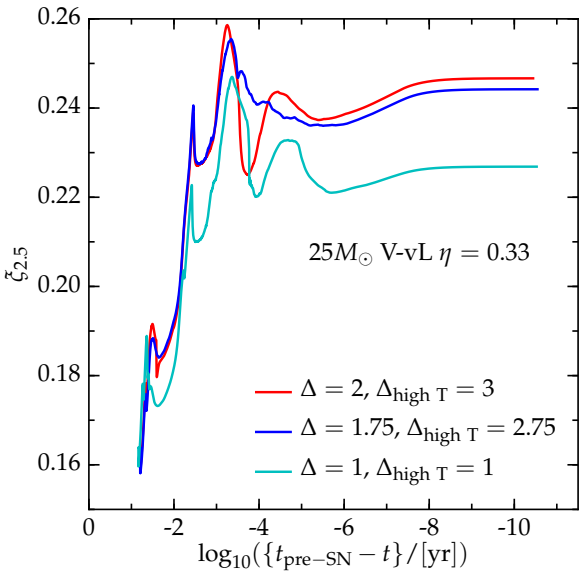

Fig. A.3: Variations in the late-time evolution of the compactness parameter with spatial resolution. The magnitude of these variations is smaller than the variations caused by uncertainties in the wind mass loss. The curves start at oxygen depletion (after the re-meshing described in Appendix A.2.1) and end at the onset of core collapse. They correspond to $25\,M_\odot$ models computed with a 203-isotope nuclear reaction network, using the V-vL algorithm and $\eta = 0.33$ and varying $\Delta \equiv$ `mesh_delta_coeff`, and $\Delta_{\mathrm{high\ T}} \equiv$ `mesh_delta_coeff_for_highT` (higher numbers correspond to lower resolution, see text). The highest resolution model (cyan curve) is the model described in Sec. 2.3.7.

Finally, an important question to address is the sensitivity to numerical resolution of the subsequent evolution toward core collapse and of the final pre-collapse structure and compactness parameter. Due to the memory limitations and computational difficulties described in the above, we are unable to carry out a rigorous convergence test with `MESA` at this time. However, in order to gain some insight into the effects of resolution, we choose the $25\,M_\odot$ V-vL $\eta = 0.33$ model and carry out two additional simulations at *reduced* resolution from oxygen depletion to the onset of core collapse using our 203-isotope nuclear reaction network. Specifically, we choose (`mesh_delta_coeff`, `mesh_delta_coeff_for_highT`) = (1.75, 2.75) and (2, 3), whereas the standard setting for our models discussed in Sec. 2.3.7 is (1, 1). Figure A.3 shows the evolution of the compactness parameter from oxygen depletion (after re-meshing) to the onset of core collapse for these models. We find that these lower-resolution models evolve qualitatively very similar to our standard model, but produce variations in the pre-SN compactness parameter of about $\sim$ 9%, which is smaller than the variations of $\sim$ 30% caused by different wind mass loss algorithm combinations.